VARONA: Verifiable Analysis of Regulatory Objects, Narratives, and Actors

An application to the Arizona Energy Promise Taskforce Report to Governor Katie Hobbs

Draft v1.0 | 2026-08-18 | Project Brief

Edwin Alvarado-Mena, PhD

Policy Data Science Consulting LLC

AlvaradoCSS.com ↗ | PolicyDataScience.com ↗

1 What VARONA does

VARONA (Verifiable Analysis of Regulatory Objects, Narratives, and Actors) is an ontology-guided system for recovering structured policy information from unstructured documents. Its central analytical unit is the **policy proposition**: a discrete assertion that expresses a policy-relevant condition, action, relationship, position, or consequence and identifies how relevant entities participate in that assertion.

The system is designed for situations in which actors need to make the most of the source material available to them, whether that means a large document collection or a small number of information-dense files. Rather than reducing documents to topics, keywords, named entities, or co-occurrence, VARONA attempts to preserve the proposition that establishes a policy relationship and the evidence from which that representation was derived (Grimmer, Roberts, and Stewart 2022; Liu et al. 2020).

The current implementation is tailored to **data-center policy and regulatory readiness**. Its substantive policy domains include energy and grid systems, water and cooling, siting and permitting, enabling infrastructure, economics and workforce, and community and environmental impacts. Cross-cutting ontology dimensions separately represent institutional roles, distributional consequences, implementation, stakeholder positions, evidence, and uncertainty. The architecture is designed so that a different domain-specific ontology can replace this implementation without changing VARONA's proposition-centered logic.

2 How the current pipeline works

VARONA currently uses a six-stage workflow. Stages 01-04 parse the source document, screen policy-relevant material, extract entities, and normalize those entities into a canonical registry. Stage 05 is a parallel policy-information recovery product rather than a serial prerequisite for proposition extraction. Stage 06 builds proposition-centered structured data from reconstructed source text together with the Stage-04 normalized entity registry.

Stage 06 is internally divided into two passes. **06A identifies policy propositions** and owns proposition admission. It works from analysis records prepared deterministically from the parsed source, including conservative reconstruction of source text split across PDF pages. **06B structures every proposition admitted by 06A**;

it does not apply a second recall gate. For each proposition, it assigns proposition semantics and participant roles using paragraph-local canonical entity IDs.

The implementation deliberately separates model-based interpretation from operations that can be performed reliably in code. Python handles reconstruction, auditing, proposition IDs, entity-surface resolution, batching, caching, schema validation, canonical entity hydration, restoration of authoritative evidence and provenance, role/type compatibility checks, and final assembly. The language model is reserved for proposition identification and semantic interpretation (Brach, Košťál, and Ries 2025; Ziems et al. 2024).

The current API policy is cost-aware. Stage 06A uses **GPT-5.6** with low reasoning, while Stage 06B uses **GPT-5.4 nano** with low reasoning. Compatible item-level results are cached and reused. Before any new paid requests are made, the runner reports the number of reusable items, new items, planned requests, and an estimated additional cost, and requires approval. The current defaults batch up to 25 analysis records in 06A and 50 proposition packets in 06B, subject to serialized-size limits. An optional stronger-model escalation path exists but is disabled by default.

3 What VARONA represents

The canonical Stage-06 output distinguishes **proposition nodes** from **entity nodes**. Entities participate in propositions through typed semantic roles. VARONA keeps four label systems distinct but aligned: natural-language labels for readable prose, canonical-schema labels for storage and validation, ontology-role labels for semantic meaning, and verb-like relational predicates for directed graph edges. Thus a **beneficiary** is stored under `BENEFICIARIES`, occupies the ontology role `BENEFICIARY`, and is displayed as an entity that *benefits from* a proposition.

Table 1: VARONA relationship labels across prose, schema, ontology, and interactive graph display.

Natural-language label	Canonical-schema label	Ontology-role label	Relational predicate
Sources	`SOURCE`	`SOURCE`	*is source of*
Policy instruments	`INSTRUMENT`	`INSTRUMENT`	*is used in*
Authorities	`AUTHORITY`	`AUTHORITY`	*has authority over*
Responsible actors	`RESPONSIBLE_ACTOR`	`RESPONSIBLE_ACTOR`	*is responsible for*
Affected actors	`AFFECTED_ACTORS`	`AFFECTED_ACTOR`	*is affected by*
Beneficiaries	`BENEFICIARIES`	`BENEFICIARY`	*benefits from*
Cost bearers	`COST_BEARERS`	`COST_BEARER`	*bears costs of*
Risk bearers	`RISK_BEARERS`	`RISK_BEARER`	*bears risk from*
Locations	`LOCATION`	`LOCATION`	*is location of*

The executive summary uses natural-language labels when discussing substantive findings and the technical labels when the distinction matters. All four forms refer to the same proposition-mediated relations.

Information that characterizes the proposition itself remains attached to the proposition rather than being expanded into unnecessary graph nodes. This includes a compact, source-faithful **predicate**, an optional controlled **action**, proposition type, status, modality, position, expected and adverse effects, evidence basis, confidence, and provenance. Separating `predicate` from `action` is important: descriptive claims can preserve what the source actually says without being forced into an inappropriate controlled policy verb.

Every proposition remains linked to verbatim evidence and document provenance (Törnberg 2024; Wang et al. 2024). Stage-06 evidence and provenance are treated as authoritative upstream information and restored deterministically during hydration rather than trusting the model to reproduce them exactly. Canonical entity names and types likewise come from the normalized entity registry, not from free-form model generation.

4 Current analytical product

The primary user-facing output is an **interactive proposition-centered knowledge network**. The network supports search, filtering, multiple analytical views, directed and labeled semantic edges, proposition inspection, source-document context, confidence cues, and inspection of predicate, controlled action, participant roles, effects, evidence, and provenance. The same network is written both as a local development artifact and as a deployment-ready `index.html`.

The renderer is downstream from Stage 06: it consumes the canonical proposition file rather than regenerating upstream analytical artifacts. R builds the visualization, while NetworkX/SciPy provide layout through a separate reticulate-managed Python environment with declared version constraints. This keeps visualization dependencies separate from the project-local Python environment used by the OpenAI API pipeline.

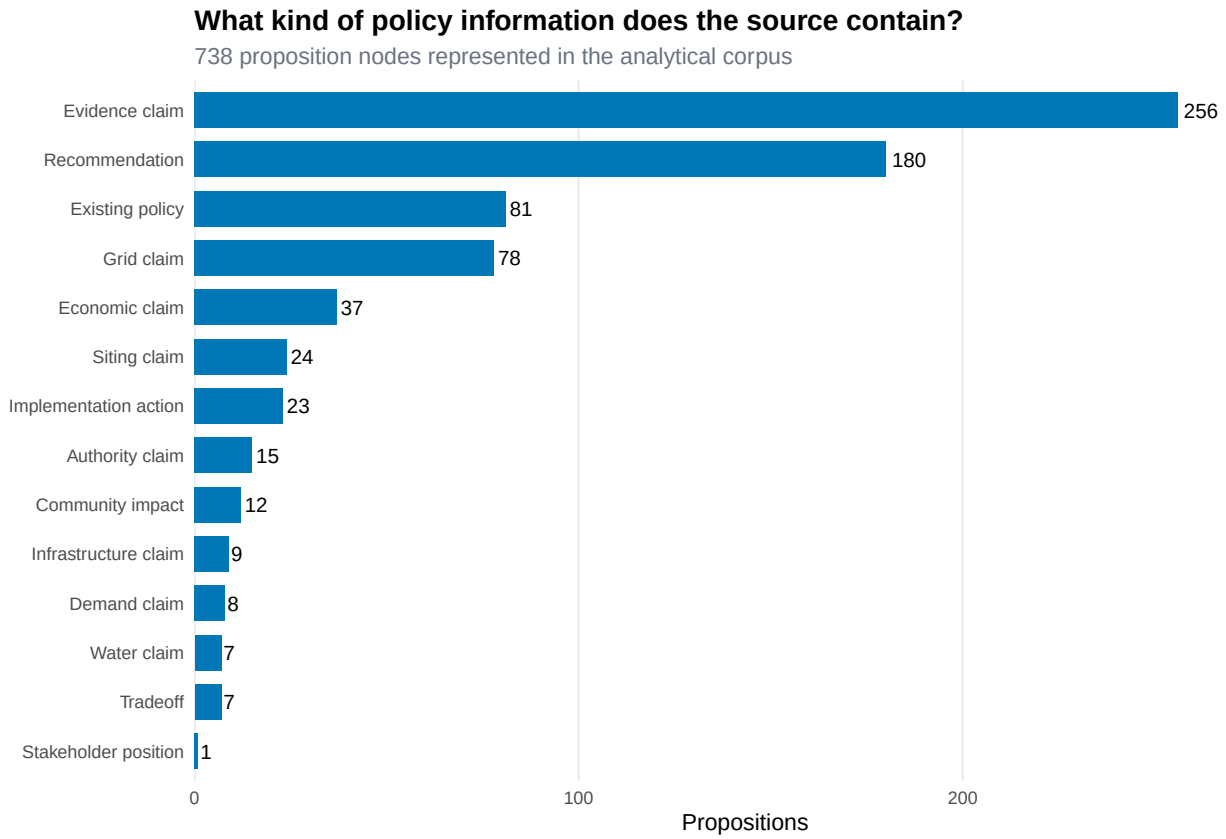
The structured proposition data also supports analytical projections focused on institutional governance, distributional effects, stakeholder politics, evidence and uncertainty, and substantive policy domains. Automated analytical reports are a natural downstream use of the representation, but the current implementation's principal delivered products are the canonical structured data and interactive network.

5 Five immediately useful visualizations

The five figures below are **generated directly from the current canonical Stage-06 output when this document is rendered**. They are not mock-ups, and producing them requires no additional model calls. To keep the executive summary aligned with the interactive network, the figures exclude the table of contents, letter, acknowledgments, and reference records.

5.1 Policy proposition profile

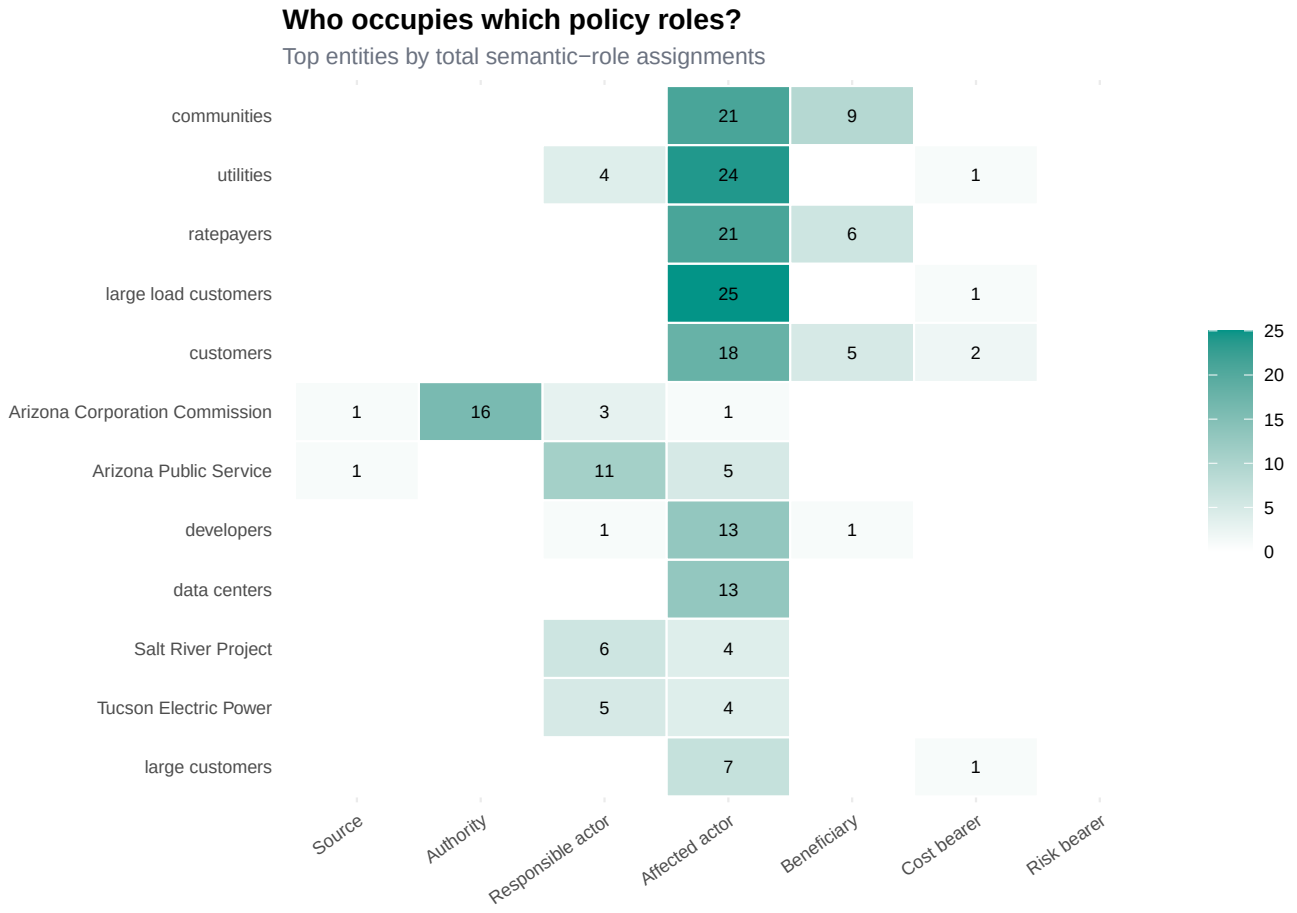
Figure 1: Policy proposition profile. Counts are generated from the current analytical Stage-06 proposition set.



Use. This is the quickest overview of the document's substantive structure. It distinguishes recommendations and existing policies from evidentiary, grid, economic, siting, authority, and other claim types instead of treating all extracted statements alike.

5.2 Actor-role map

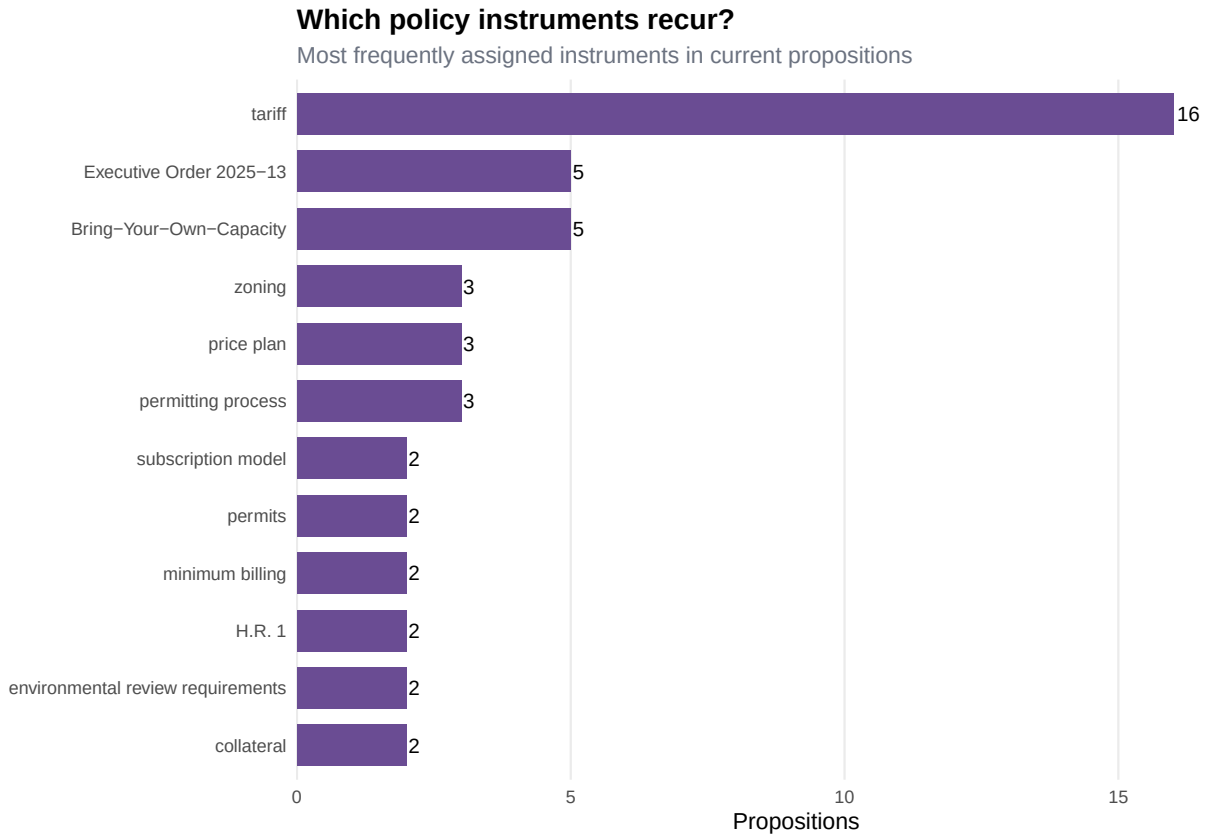
Figure 2: Actor-role map. Cell values count proposition-mediated role assignments for the most frequently represented entities.



Use. The same actor can appear as an authority in one proposition, an implementer in another, and an affected party elsewhere. This view exposes those differences directly rather than converting repeated mentions into generic ties.

5.3 Policy-instrument landscape

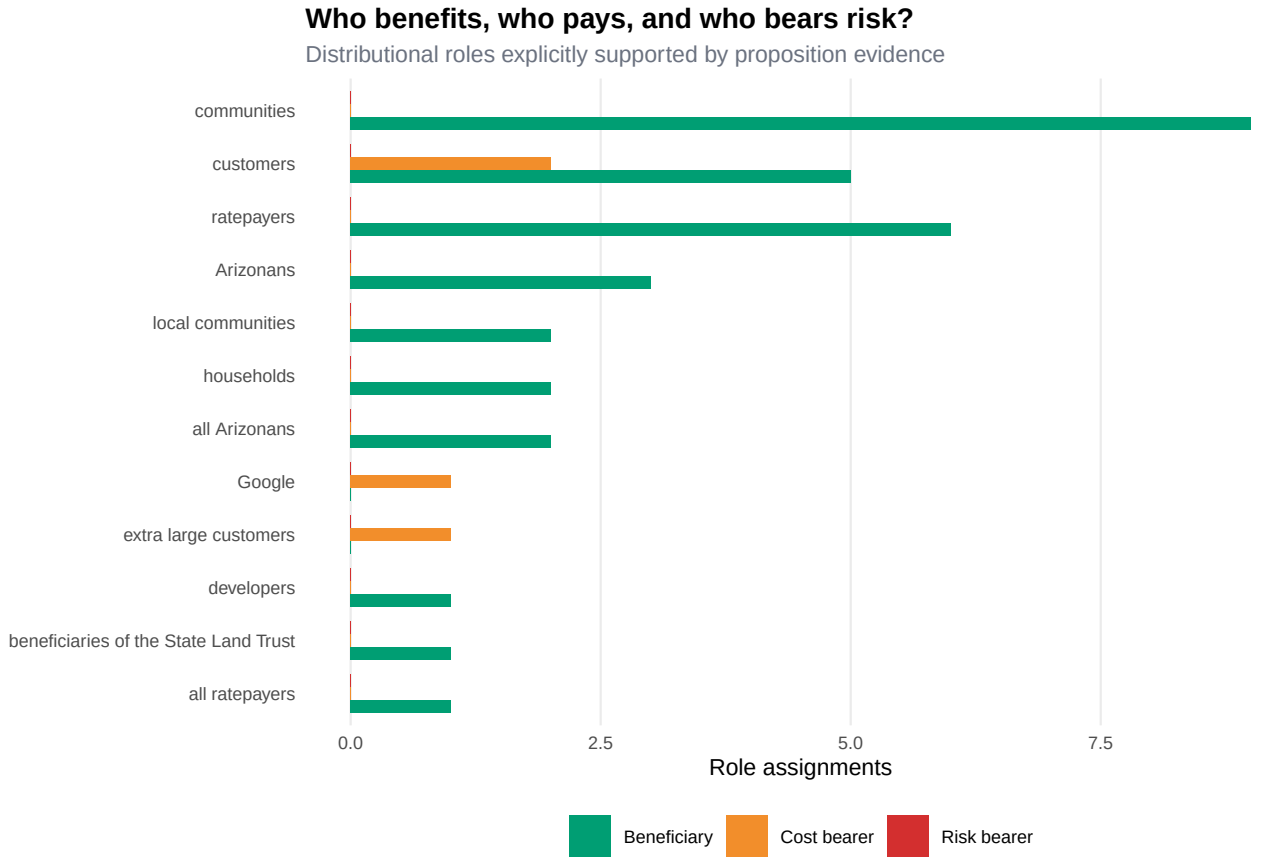
Figure 3: Policy-instrument landscape. The chart ranks instruments by the number of distinct propositions in which each is assigned the instrument role.



Use. This view moves from *what the document discusses* to *how policy is actually operationalized*. It can quickly surface the mechanisms around which implementation and regulatory strategy are concentrated.

5.4 Distributional impact view

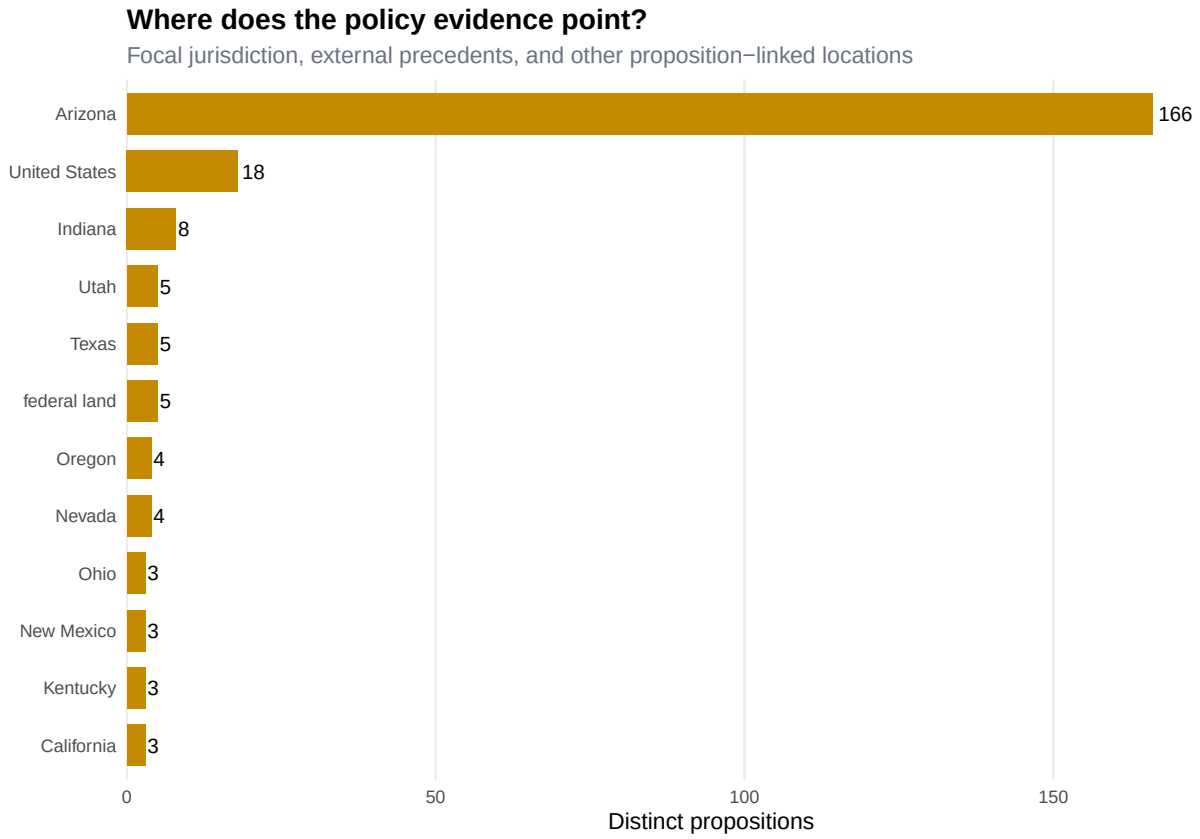
Figure 4: Distributional impact view. Counts distinguish entities explicitly represented as beneficiaries, cost bearers, or risk bearers.



Use. This is an immediate distributional-policy lens. It separates favorable, costly, and risky positions instead of treating all affected entities as equivalent.

5.5 Geographic policy footprint

Figure 5: Geographic policy footprint. The ranked view counts distinct propositions linked to each location or jurisdiction.



Use. The source is Arizona-focused, but it also invokes external jurisdictions and examples. This view makes the geographic reach of those propositions immediately visible and provides a starting point for identifying comparative precedents.

Together, these figures demonstrate a central property of VARONA: a visualization is an **entry point into evidence**, not a detached summary. Each count originates in structured propositions that remain linked to verbatim evidence and document provenance; the interactive network can then be used to inspect the propositions behind the pattern.

6 Why the implementation matters

VARONA's value is not simply that it uses an LLM to read policy documents. Its contribution is the **combination of ontology-guided semantic extraction, deterministic controls, evidence preservation, and inspectable graph representation**. The model performs bounded interpretive tasks; code enforces what can be enforced mechanically; and users retain access to the source material behind every structured claim.

This design makes the system useful for policy environments in which accuracy, traceability, and analytical structure matter as much as retrieval speed. It provides a foundation for regulatory monitoring, institutional and distributional analysis, stakeholder intelligence, comparison across documents or jurisdictions, and reproducible policy reporting while keeping human inspection possible at the proposition level.

Brach, William, Kristián Košt'ál, and Michal Ries. 2025. "The Effectiveness of Large Language Models in Transforming Unstructured Text to Standardized Formats." *IEEE Access*.

Grimmer, Justin, Margaret E Roberts, and Brandon M Stewart. 2022. *Text as Data: A New Framework for Machine Learning and the Social Sciences*. Princeton University Press.

Liu, Kang, Yubo Chen, Jian Liu, Xinyu Zuo, and Jun Zhao. 2020. "Extracting Events and Their Relations from Texts: A Survey on Recent Research Progress and Challenges." *AI Open* 1: 22–39.

Törnberg, Petter. 2024. "Best Practices for Text Annotation with Large Language Models." *arXiv Preprint arXiv:2402.05129*.

Wang, Xinru, Hannah Kim, Sajjadur Rahman, Kushan Mitra, and Zhengjie Miao. 2024. "Human-LLM Collaborative Annotation Through Effective Verification of LLM Labels." In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 1–21.

Ziems, Caleb, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. "Can Large Language Models Transform Computational Social Science?" *Computational Linguistics* 50 (1): 237–91.